

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium



13 - 17 September 2010

Crossing Borders within the **ABC**

Automation,

Biomedical Engineering and

Computer Science



Faculty of
Computer Science and Automation

www.tu-ilmenau.de

th
TECHNISCHE UNIVERSITÄT
ILMENAU

Home / Index:

<http://www.db-thueringen.de/servlets/DocumentServlet?id=16739>

Impressum Published by

Publisher: Rector of the Ilmenau University of Technology
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff

Editor: Marketing Department (Phone: +49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

Faculty of Computer Science and Automation
(Phone: +49 3677 69-2860)
Univ.-Prof. Dr.-Ing. habil. Jens Haueisen

Editorial Deadline: 20. August 2010

Implementation: Ilmenau University of Technology
Felix Böckelmann
Philipp Schmidt

USB-Flash-Version.

Publishing House: Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilmenau

Production: CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts

Order trough: Marketing Department (+49 3677 69-2520)
Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher: Universitätsbibliothek Ilmenau
[ilmedia](#)
Postfach 10 05 65
98684 Ilmenau

© Ilmenau University of Technology (Thür.) 2010

The content of the USB-Flash and online-documents are copyright protected by law.
Der Inhalt des USB-Flash und die Online-Dokumente sind urheberrechtlich geschützt.

Home / Index:

<http://www.db-thueringen.de/servlets/DocumentServlet?id=16739>

PARTICLE SWARM INTELLIGENCE: A PARTICLE SWARM OPTIMIZER WITH ENHANCED GLOBAL SEARCH QUALITIES AND GUARANTEED CONVERGENCE

Hendrik Große-Löscher

Postgraduate, Faculty of Computer Science and Automation,
Institute for Automation and Systems Engineering, Ilmenau University of Technology

ABSTRACT

A new particle swarm optimizer is presented. Particle swarm optimization PSO is a relatively young optimization algorithm related to evolutionary computation. This metaheuristic is best suited to optimize nonlinear functions. After a brief introduction to particle swarm optimization giving insights to the underlying paradigms, advantages and drawbacks are highlighted. The operation of the canonical PSO in a synthetically generated 2D test environment with three different dynamic landscapes shows the failure of the original version in transient conditions, leading to a curious phenomenon of ‘linear collapse’. Consecutively, an advanced algorithm demonstrates the potential of PSO in dynamic applications. Based on the experience of two engineering optimization tasks, a refined optimizer using constriction coefficient strategy is introduced and compared to the classic algorithm and a later introduced version with implemented inertia weight. The new optimizer for static optimization problems incorporates superior global search characteristics and guarantees final convergence.

Index Terms – Particle swarm optimization, dynamic environments, linear collapse, constriction coefficient, inertia weight, global search, convergence

1. INTRODUCTION

Particle swarm optimization - PSO - was introduced in 1995 by Russell Eberhart and James Kennedy, an electrical engineer and a social psychologist. In recent years particle swarm intelligence has gained a lot of recognition as PSO proved to be an effective method to handle different kind of optimization problems. Meanwhile, the application areas span from engineering tasks to economics. Related to swarm intelligence and evolutionary computation, PSO is a metaheuristic whose paradigms are inspired by bird flocking and fish schooling. The precursor of the canonical PSO algorithm was originally intended as a graphical simulator of the graceful but unpredictable choreography of a flock of birds. The originators discovered the potential of the method to optimize

continuous nonlinear functions through the simulation of a simplified social milieu. PSO does not incorporate the principle ‘survival of the fittest’ like it is used in genetic algorithms. From the beginning to the end of an optimization run all particles ‘survive’ and actively search the function area for optima. A particle is a tuple of all parameters and thus contains a possibility for solving a given function. After an initial distribution within the search area which is often random, the particles move through the search space with a direction-dependent velocity. After each movement the fitness of a particle is evaluated. Fitness equals the numerical function value where the particle is located. As the swarm members interact with each other based on the information of the best solution of the entire swarm found so far (global best) and the best individual location determined (personal best), cognitive/perception-based and social patterns can be derived. The particles are attracted by the best positions and move towards the global best. During the optimization process the particles may encounter positions with higher fitness values. Depending on the chosen PSO variant and the adjusted parameters, the global search ability, swarm behaviour, convergence rate and the quality of the final solution are basically influenced. The algorithm proved to be very simple, robust and highly efficient. Regarding the dimensionality and the complexity of the optimization task no limitations or restrictions occur.

The outline of the paper is as follows. Section 2 briefly explains the basic principles of PSO. Following the scheme for static optimization tasks, a special focus is put on dynamic environments with three different kind of optimization problems (height change of peaks, location change of peaks with constant heights and location change of peaks with changing heights). The application of the canonical PSO to a synthetically generated dynamic 2D test case demonstrates the failure of the algorithm to track changing peaks, leading to a remarkable appearance of ‘linear collapse’. An implementation of an advanced PSO technique denotes its feasibility in changing conditions. A distinguished PSO variant is described in Section 3. After the recapitulation of the commonly used constriction coefficient limiting the particles’ velocities, a refined algorithm with a simple

but effective new strategy is introduced. The implementation leads to a significant improvement of the global search ability and enables an enhanced control of the swarm's convergence behaviour at the end of an optimization run. Consequently, the excellence of the final result is influenced and the swarm can be adapted to the optimization task. Experimental investigations are illustrated and indicate the outstanding performance of the new approach. Section 4 briefly specifies the application of PSO to two engineering tasks with 11 and 361 parameters respectively. Finalizing the work, section 5 summarizes the results.

2. PARTICLE SWARM OPTIMIZATION

Optimization aims to find the minimum or maximum of an objective function within a predefined search area. Beside other mathematical scopes, this work is focused on nonlinear function optimization including constraints. The objective function can either be single-objective or multi-objective, e.g. a function with standardized weighted objectives where constraints are implemented by penalty terms and the problem is converted into a single objective function.

2.1. Static environments

PSO is a stochastic, gradient-less and derivative-free nature-analogous algorithm that is based on a set of particles. A particle includes a tuple of all parameters of the optimization task and thus provides a potential solution. After the initial distribution of the particles within the search space, all particles attain direction-dependent and dimension-individual velocities. The operational sequence for *static applications* is as follows [4, 5, 6].

1. Stochastic initialization of the particle population within the function/search area.
2. Calculation of the fitness of each particle.
3. Modification of the individual velocity based on the best individual and best global position so far (neighbourhood).
4. Determination of the new positions of the particles.
5. Fitness evaluation (2.) of each particle, with convergence/termination criterion: END, otherwise go to 3.

The referring equation for the velocity update is

$$v_{id}(t) = v_{id}(t-1) + c_1 \cdot \text{rand}(\) \cdot (p_{id} - x_{id}(t-1)) + \dots \\ \dots + c_2 \cdot \text{Rand}(\) \cdot (p_{gd} - x_{id}(t-1)) \quad (1)$$

The new positions are updated by

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t) \quad (2)$$

Equations (1) and (2) employ the denotations

x_{id} : potential solution, location of particle i in dimension d
 v_{id} : velocity of particle i in dimension d
 p_{id} : best location so far of particle i in dimension d
 p_{gd} : best location so far of the best particle g of all neighbours of particle i in dimension d
 c_1 : cognitive parameter/acceleration constant
 c_2 : social parameter/acceleration constant
 $\text{rand}()$, $\text{Rand}()$: equally dispersed random numbers from $[0,1]$

In equation (1), t is the current iteration step. The movement of the particles following equation (1) is based on a so called cognitive term (term 2) and a social term (term 3) along with the velocity $v_{id}(t-1)$ (term 1) which is equal to a momentum. The cognitive constant c_1 influences the individual particle behaviour regarding its own best position. The social constant c_2 controls the movement towards the direction of the particle which currently has the best position in the swarm. c_2 influences the behaviour of particle i with reference to the fitness of the neighbours and thus describes a component of social behaviour. In each iteration, the cognitive and social part of the movement is varied by random to maintain diversity. During their move, particles can find locations which are characterized by a higher fitness than the best optimum found so far. The behaviour of the swarm is defined by the experience of each individual swarm member, its current position and the exchange of information by the individual swarm members among themselves (orientation of the individuals to group orders). Thereby, PSO successfully imitates the natural behaviour of animals in groups or swarms.

The swarm size, neighbourhood size and topology affect the swarm behaviour and thus influence the search characteristics. Beside several approaches, two main formulations concerning the particles' communication topology exist, namely *gbest* and *lbest*. In the *gbest* model, each particle is influenced by the best particle of the entire swarm while in the *lbest* model each particle is influenced by the particles of the local neighbourhood. In many applications, the *gbest* model tends to converge faster.

Another mentionable topic is the information analysis of the best position. In a so called synchronous PSO which the original version is related to, the best positions are updated after all particle movements in one iteration step. The asynchronous PSO updates the best positions after each particle movement which allows an immediate feedback about the best regions and leads to a higher convergence rate. Based on application experiences with the canonical PSO, numerous optimization runs with small populations are more effective in finding an admissible solution than few runs with large populations. This surprising effect is based on the fast convergence of the PSO.

2.2. Dynamic environments

Based on the belief that PSO converges fast, the dynamic tracking of peaks seemed to be possible [10]. Today, with deeper insights into PSO and many applications at hand, the conventional PSO is inapplicable to dynamic environments. Due to its immanent paradigms, PSO fails to most dynamic environments.

A synthetical 2D test environment mainly consisting of trigonometrical expressions is generated to provide three different dynamic test cases:

$$y(x_1, x_2) = [(-x_1^2 - x_2^2) \cdot \sin(\frac{x_1}{4}) \cdot 2 \cdot x_1 \cdot x_2 \cdot \dots \quad (3)$$

$$\dots \cdot \csc h(x_1) \cdot \csc h(x_2) + \sin(3 \cdot x_1) + \sin(3 \cdot x_2)] \cdot \dots$$

$$\dots \cdot \sin(3 \cdot x_2) + \sin(3 \cdot x_1)] \cdot \cos(x_1) \cdot \sin(x_2) \cdot \tan(x_1)$$

with $x_1, x_2 \in [-5.5; 5.5]$

This unsimplified equation provides the following search area.

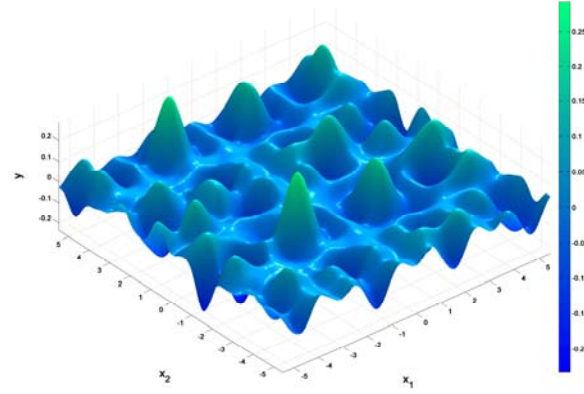


Figure 1: 2D test environment

To develop a proper test suite, three different dynamic tasks are included:

- height change of peaks (α)
- location change of peaks with constant heights (β)
- location change of peaks with changing heights (δ)

During all calculations, 155 iterations are performed in one way and another 155 iterations in the other way so that the search area moves forward and backward. A sidewise slide with constant heights (β) equals $\Delta x_1 = 0.05$ while $\Delta x_1 = 0.02$ when the peaks also encounter a change in heights (δ). Δx_2 is constantly kept to zero. Two additional terms in equation (3) ensure the increasing and decreasing heights of the peaks.

The application of the conventional algorithm to the three different dynamic tasks indicate the failure of the PSO technique to identify and trace single peaks as well as to find and trace the global optimum. The main deficit of the classic PSO version in dynamic environments is the *outdated memory* of the particles when a change in the function occurs additionally to a *fatal loss of diversity*. Attuned with standard parameter settings, the PSO rapidly converges to a

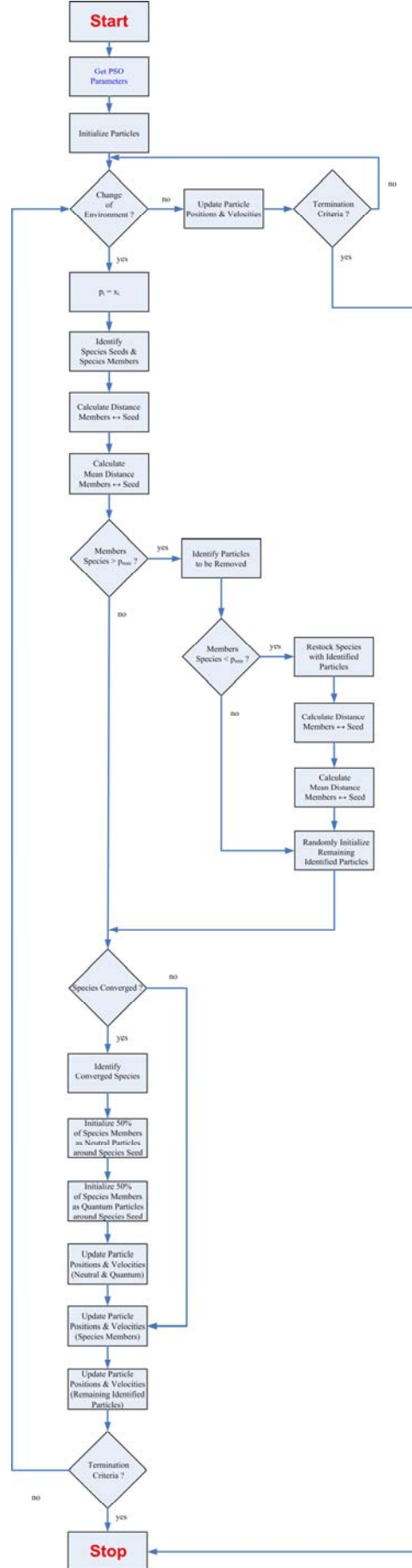


Figure 2: flow diagram of PSO for dynamic tasks

single peak and only performs a weak local search around the peak (test case α and δ). The parameters are $c_1=c_2=2.05$, as commonly applied [4, 6], v_{\max} (the maximum move of a particle in one dimension) is set to 0.5 and if a particle tends to leave the search space, it is placed to the limit value of the dimension restricting the search area. Soon after starting test instance β , all particles are collapsing to a single line, showing a curious phenomenon called ‘linear collapse’ [2]. The length of the line is almost constant for approximately two thirds of the iterations and amounts to $\Delta x_2=1.44$.

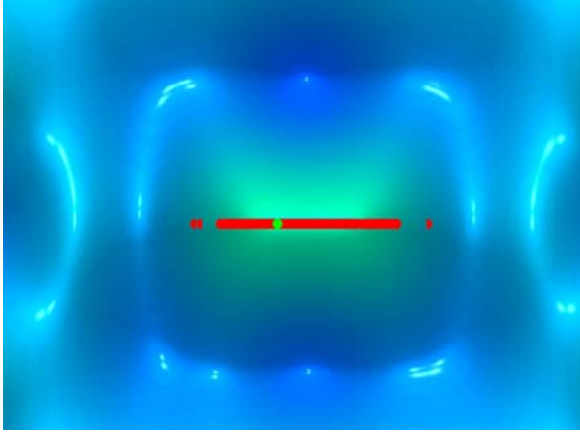


Figure 3: ‘linear collapse’ as seen from above

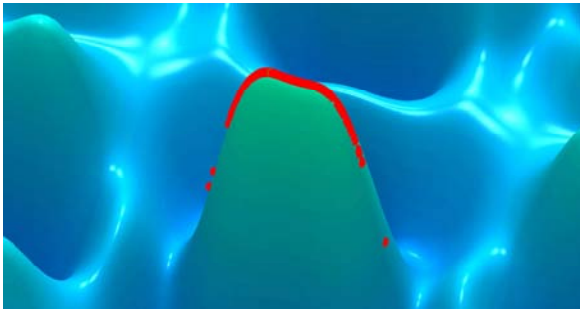


Figure 4: ‘linear collapse’ laterally seen

As the phenomenon appears, all velocities in direction of x_1 permanently receive the maximum velocity $-v_{\max,1}=-0.5$, while the velocities in direction of x_2 are alternating between $\pm v_{\max,2}=\pm 0.5$. Animations and the analysis of the velocities indicate the establishment of alternating linear particle trajectories. While $v_1=\text{const.}$, all particles are moving between the best particles/attractors which are mostly at the both ends of the path.

The utilisation of PSO for dynamic environments still is a young field of activity. Figure 2 shows the flow diagram of a proposed algorithm for dynamic tasks which is mainly based on [3]. Without going into detail, the foremost principles are the *permanent update of the best positions*, the *identification of species seeds* and their members with a predefined amount of minimum and maximum particles and the *initialisation of neutral and quantum particles* around

the species’ seed. These particles are initialised when the species is converged to achieve a balanced ratio of convergence and diversity. The subsequent results indicate the performance of the quoted algorithm.

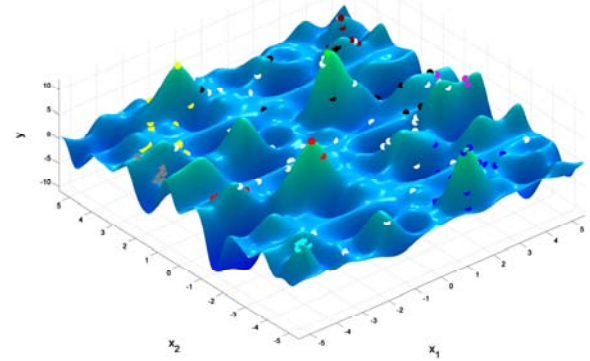


Figure 5: PSO applied to test case α

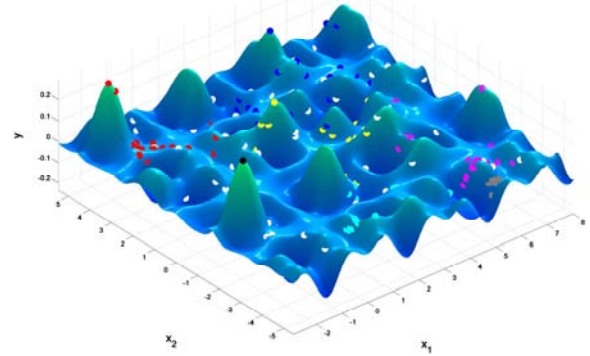


Figure 6: PSO in dynamic test case β

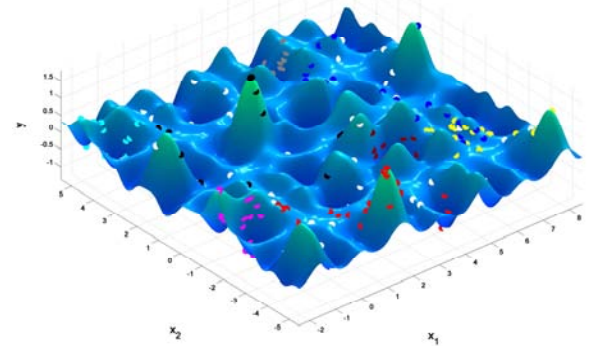


Figure 7: PSO in dynamic environment δ

All procedures employ 200 particles. The radius of a species is set to $r_{\text{species}}=4$ around a seed, the minimum amount of particles per species is 15 while 35 particles are the maximum quantity. To initialize the neutral and quantum particles around a species’ seed, a radius of $r_{\text{cloud}}=0.2$ is applied. The convergence threshold determining that a species is converged is set to $\Delta=0.0001$. Δ correlates to a mean distance between a species and the species’ seed. Neutral particles provide convergence while the randomly initialized quantum particles within r_{cloud} ensure diversity of the species. The current PSO version is capable of detecting and tracing all major dynamic peaks. The algorithm and its success seriously depend on the problem-specific parameter settings. Future work is encouraged to generalize this tuning.

3. PSO WITH CONSTRICTION COEFFICIENT

Numerous works are dedicated to improve PSO and its premature convergence. Most versions study the swarm behaviour by employing different *inertia weight* (adding a factor to term 1 in equation (1)) and *constriction coefficient* approaches. The aim is to control both, *exploration* and *exploitation*, *global* and *local search*.

3.1. Common constriction coefficient strategy

Mathematically, the implementation of a constriction coefficient is a special case of the inertia weight version. The standard constriction coefficient algorithm [1, 4, 6, 10] is an extension of equation (1).

$$v_{id}(t) = \chi \cdot [v_{id}(t-1) + c_1 \cdot rand() \cdot (p_{id} - x_{id}(t-1)) + \dots + c_2 \cdot Rand() \cdot (p_{gd} - x_{id}(t-1))] \quad (4)$$

According to [4], χ is defined as

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}} \quad \text{with } \varphi = c_1 + c_2, \varphi > 4 \quad (5)$$

Commonly, $c_1 = c_2 = 2.05$ resulting in $\chi = 0.72984$.

3.2. Refined constriction coefficient strategy

Based on application experiences [7, 8, 9], the refined utilisation of χ can significantly improve the swarm behaviour.

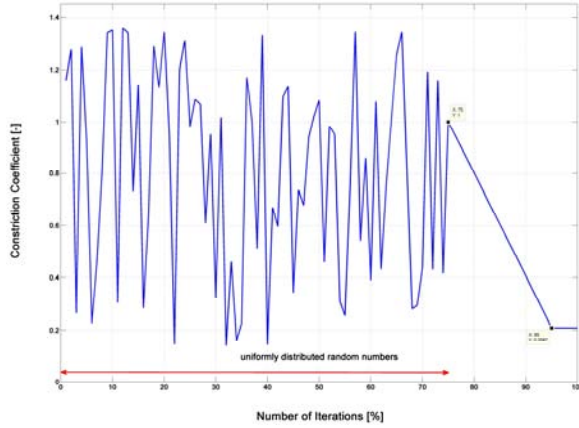


Figure 8: exemplary trend of χ

The simple but effective procedure consists of uniformly distributed random numbers within the first 75% of the iterations in a proposed range of at least [0.1;1.3]. Between 75% and 95% of the iterations, χ decreases linearly from 1 to 0.20871 which results of equation (5) with $c_1=2$ and $c_2=5$. Within the last 5% of the iterations, χ is kept constant at 0.20871. An innovative implementation of a random $\chi > 1$ leads to a *significant improvement of the global search ability* (exploration). The idea is to provide an *exhaustive global search* that *maintains diversity* which is

followed by an *intensive local search* (exploitation) with *controlled convergence*, *suitable for many tasks*.

3.3. Experimental results

The sequences below demonstrate the various swarm behaviours of different PSO versions in the static 2D test case of equation (3).

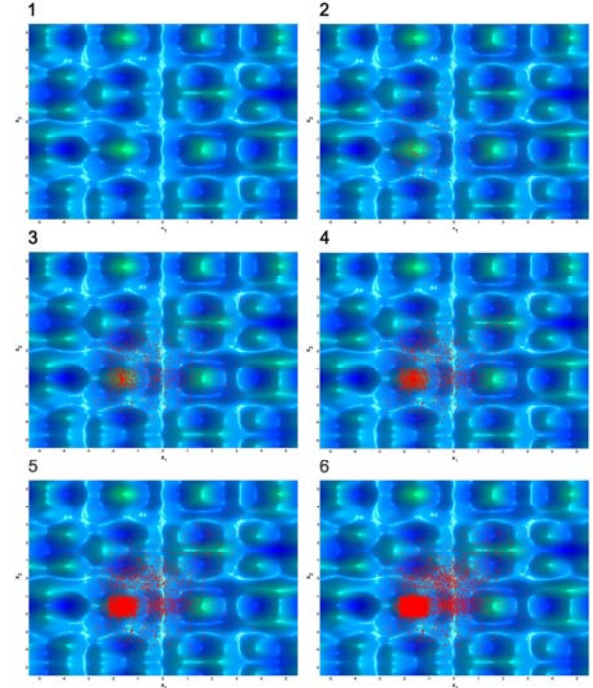


Figure 9: positions visited by conventional PSO

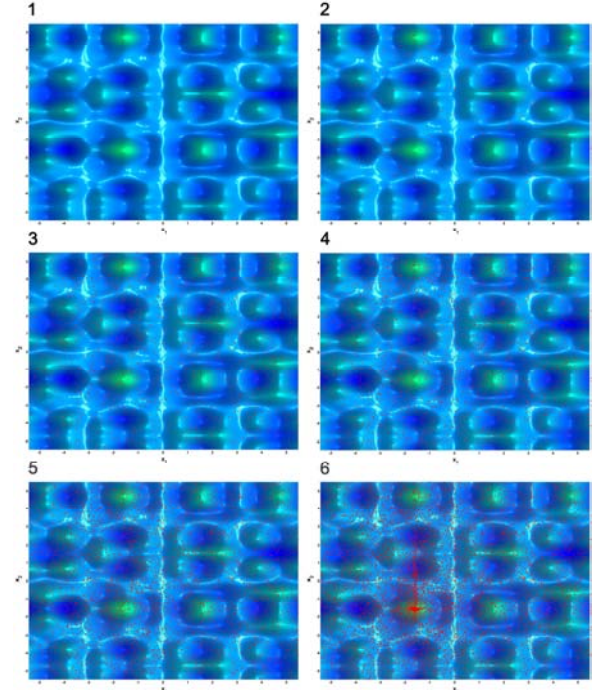


Figure 10: positions visited by PSO with constriction

The figures show all visited positions of the PSO with 75 particles on their way of 200 iterations at 1 (1), 10 (2), 30 (3), 50 (4), 100 (5) and 200 iterations (6), seen from above. The conventional PSO and the version

with inertia weight (not presented) have a quite similar performance and show premature convergence with a weak global search. The conventional constriction version keeps χ constant and incorporates an early convergence after one third of the iterations with no further global search. The refined constriction algorithm has a superior global search capability followed by a controlled local search. The conventional version as well as the algorithm with a linear decreasing inertia weight from 0.9 to 0.4 [6, 10] utilize $c_1=c_2=2.05$, $v_{\max}=0.5$ and set the particles to the border line when they tend to leave the search area. An improved version of [1] randomly varies χ within $[0.2;0.9]$ over all iterations. The refined constriction algorithm employs the presented trend of χ , but within a range of $[0.1;1.7]$. The choice of c_1 and c_2 results of the desire to boost global search, but as animations indicate, the strategy *massively softens the swarm characteristics and solidarity inside the random phase of χ* so that their proper selection becomes less important. The search seems to become *fully random*. When particles desire to leave the search area, they are stochastically reinitialized. This is an essential mechanism, as otherwise particles would only sit on the borderline without active search. A great improvement is thus the *discontinuation of the sensitive parameter v_{\max} , reducing the parameters*. The upper bound of the constriction coefficient should be at least 1.5 to insure global search, while the lower bound is less important. Experiments with higher values (up to 30) do not show significant changes in the swarm behaviour for the proposed task. The final value of χ is sensitive. Is the value too low, particles might be slowed down too much so that they cannot perform a local search, while an excessive χ allows steps that are too vast so that the convergence is weak.

4. ENGINEERING APPLICATIONS

The constriction PSO as described has been applied to several engineering applications with prosperous results.

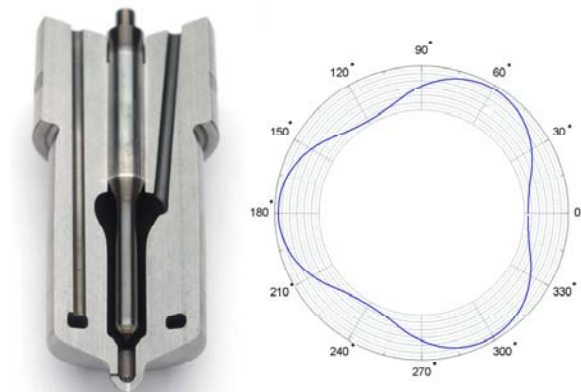


Figure 11: optimization of injection nozzle and cam

The optimizations concern 11 parameters, 47 constraints, 23 equations and 6 objective criterions

(nozzle) as well as up to 361 parameters, 7 constraints, 86 equations and 6 objective criterions (cam) [7, 8, 9].

5. CONCLUSIONS

A new PSO algorithm with a refined constriction coefficient strategy is presented. The algorithm provides superior global search quality and guarantees convergence to balance exploration and exploitation.

6. REFERENCES

- [1] Achtnig, J.: Particle Swarm Optimization with Mutation for High Dimensional Problems. Studies in Computational Intelligence (SCI) 82, pp. 423-439, Springer-Verlag, Berlin, 2008
- [2] Bentley, P., Blackwell, T.: Dynamic Search with Charged Swarms. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '02), pp. 19-26, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002
- [3] Blackwell, T., Branke, J., Li, X.: Particle Swarm with Speciation and Adaptation in a Dynamic Environment. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '06), pp. 51-58, ACM, New York, NY, 2006
- [4] Clerc, M.: Particle Swarm Optimization. ISTE - Hermes Science Publishing, London, 2006
- [5] Eberhart, R., Kennedy, J.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, IV, pp. 1942-1948, Piscataway, NJ, 1995
- [6] Eberhart, R., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, 2001
- [7] Große-Löscher, H., Haberland, H., Yalcin, H.: Verfahren zur Optimierung einer Einspritzdüse für eine Brennkraftmaschine. Offenlegungsschrift, Deutsches Patent- und Markenamt, DE 102006043460 A1 2008.03.27, München, 2008
- [8] Große-Löscher, H., Haberland, H.: Schwarmintelligenz zur Optimierung von Einspritzdüsen. MTZ - Motortechnische Zeitschrift, Nr. 2, S. 80-85, Springer Automotive Media, Wiesbaden, 2010
- [9] Große-Löscher, H.: Application of PSO for the optimization of Diesel engine operation. Deliverables D11.2.b+c, HERCULES Integrated Project, EU Sixth Framework Program, TIP3-CT-2003-506676, Augsburg, 2007
- [10] Shi, Y.: Particle Swarm Optimization. Feature Article, IEEE Neural Networks Society, 2004